A Program For Multivariate Normal Bayesian Allocation

and

Examples of Wide Discrepancies for Dimension Reducing Optimal Allocation and Separation Procedures in Multivariate Normal Populations

by

Robert E. McCulloch*
University of Minnesota
Technical Report No. 421

*Work supported by NIH Grant GM 25271.

1.1 Introduction

Given a set of multivariate observations, each of which is known to have been generated by one of several multivariate normal populations, it is natural to ask from which population has each observation come. These populations are identified but their parameters are unknown.

Given (i) for each population a set of observations which are known to have come from that population (the training sample) and (ii) a prior distribution giving the probabilities that an observation is generated by a specific population, Geisser (1) using Bayesian analysis, has derived posterior probabilities for allocations of observations of unknown origin to the specified populations.

1.2 The Model

At this point we need not assume normality.

We start with a random vector $(I_1,\ldots,I_n,Z_1,\ldots,Z_n,\Theta)$. Θ is a vector of parameters. I_j is the index number giving the population from which Z_j originates, Z_j being the observed response.

The probability structure of the random vector is as follows.

(i) the $\{I_j\}$ are independent and identically distributed.
$I_j \in \{1,2,\ldots,k\}$ so that there are k identified populations.
Let $P(I_j = i) = q_i$.

(ii) θ is independent of $\{I_j\}$ and has density $g(\theta)$, $\theta \in \mathbb{R}^s$.

(iii) conditional on $\{I_j\}$ and Θ the $\{Z_j\}$ are independent. Let the density be $\prod_{j=1}^{n} f(z_j|\theta,i_j)$; $Z_j \in \mathbb{R}^P\ \forall j$.

The above structure is assumed to be known. The problem is that we do not observe Θ and the set $\{I_j\}$.

1.3 The Analysis

Now suppose we have $X = \{x_{ij}\}$, $i = 1,2,\ldots,k$, $j = 1,2,\ldots,N_i$ where x_{ij} is the j^{th} observation known to have come from population i. All of these observations are realizations of independent random variables.

Further analysis is conditional upon these observations. Let $L(X|\theta)$ be the corresponding likelihood function. Then $p(\theta|X) \propto L(X|\theta)\, g(\theta)$. The probability structure is now just as above except that $g(\theta)$ is replaced by $p(\theta|X)$.

We do not observe Θ nor are we directly interested in inference about Θ. Thus it is reasonable to integrate it out and consider the marginal distribution of $(\{I_j\},\{Z_j\})$. Thus

$$f(z_1,\ldots,z_n|X,i_1,\ldots,i_n) = \int \prod_{j=1}^{n} f(z_j|\theta,i_j)\, p(\theta|X)\, d\theta. \tag{1}$$

Having observed $Z_j = z_j$, $j = 1,2,\ldots,n$

$$P(I_1=i_1,\ldots,I_n=i_n|X,z_1,\ldots,z_n) \propto \prod_{j=1}^{n} q_{i_j} \quad f(z_1,\ldots,z_n|X,i_1,\ldots,i_n) \tag{2}$$

This is a solution to the allocation problem.

When $n = 1$ the above is called single allocation. When $2 \leqslant n < \infty$ we call it joint allocation. If interest centres on I_n we calculate the marginal distribution $P(I_n=i_n|X,z_1,\ldots,z_n)$. This is referred to as sequential allocation.

Further Analysis Applicable if the N_i are Random

Recall that N_i is the number of observation in the training sample from population i. If N_i is random we need not assume the q_i to be known (see Geisser (2), page 3). Instead we assume a prior distribution

$$G(q_1,\ldots,q_{k-1}) \propto \prod_{i=1}^{k} q_i^{\alpha i},$$

the Dirichlet density.

Then $p(q_1,\dots,q_{n-1}|N_1,\dots N_{k-1}) \propto \prod_{i=1}^{k} q_i^{N_i+\alpha_i}$, where $N_k = N - N_1 - N_2 - \dots - N_{k-1}$ and N is fixed.

Now thinking of the marginal model $(\{I_j\},\{Z_j\})$ as being conditional on $(q_1,\dots,q_{k-1})$ we may calculate the density $f(z_1,\dots,z_n|X,q_1,\dots,q_{k-1})$ $= \sum_{i_1=1}^{k} \cdots \sum_{i_n=1}^{k} q_{i_1},\dots,q_{i_n} \; f(z_1,\dots,z_n|X,i_1,\dots,i_n)$. Thus

$$p(q_1,\dots,q_{k-1}|z_1,\dots,z_n, N_1,\dots,N_{k-1}) \propto p(q_1,\dots,q_{k-1}|N_1,\dots,N_{k-1})$$

$$f(z_1,\dots,z_n|X,q_1,\dots,q_{k-1})$$

From which we obtain,

$$P[I_1=i_1,\dots,I_n=i_n|X,z_1,\dots,z_n]$$

$$= \int P[I_1=i_1,\dots,I_n=i_n|X,z_1,\dots,z_n,q_1,\dots,q_{k-1}]$$

$$p(q_1,\dots,q_{k-1}|z_1,\dots,z_n,\dots,N_{k-1}) \, dq_1,\dots,dq_{k-1}.$$

$$= \int \prod_{j=1}^{n} q_{i_j} \; f\,(z_1,\dots,z_n|X,i_1,\dots,i_n)$$

$$\prod_{j=1}^{k} q_j^{N_j+\alpha_j} \, dq_1,\dots,dq_{k-1}$$

$$= f(z_1,\dots,z_n|X,i_1,\dots,i_n) \int \prod_{j=1}^{k} q_j^{N_j+n_j+\alpha_j} \, dq_1,\dots,dq_{k-1}$$

(where n_j is the number of i_s such that $i_s = j$)

$$= f(z_1,\dots,z_n|X,i_1,\dots,i_n) \; \frac{\prod_{j=1}^{k} \Gamma(N_j+n_j+\alpha_j+1)}{\Gamma(N+n+\alpha+k)} \qquad (3)$$

where $\alpha = \sum_{j=1}^{k} \alpha_j$.

In the case $n = 1$ this reduces to

$$P[I_1 = i|X,z_1] = \frac{(N_i + \alpha_i + 1)\; f(z|X,I_i = i)}{\sum_{j=1}^{k} (N_i + \alpha_j + 1)\; f(z|X,I_i = j)} \;.$$

1.4 Multivariate Normal Populations

Each population is now p dimensional normal with mean μ_i and covariance Σ_i, $i = 1,2,\dots,k$. These parameters are unknown. The analysis is carried out with and without the assumption that the Σ_i are all equal. The details of the computation will be given only in the case of joint allocation with the Σ_i not known to be equal as the method is similar in all cases.

Notation

(i) $x_{ij} \in \mathbb{R}^p$ is the j^{th} observations from the i^{th} population in the training sample $i = 1,2,\dots,k$, $j = 1,2,\dots,N_i$.

(ii) $\bar{x}_i = \sum_{j=1}^{N_i} x_{ij}/N_i$

(iii) $(N_i - 1)S_i = \sum_{j=1}^{N_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T$

(iv) $(N - k)S = \sum_{i=1}^{k} (N_i - 1)S_i$

(v) $N = \sum_{i=1}^{k} N_i$

(vi) in the case of joint allocation, let M_i be the $p \times n_i$ matrix of observations assumed to be from population i. $\sum_{i=1}^{k} n_i = n$.

In each of the following cases we give (1) above. This result may be used with either (2) or (3).

Equal Covariances

Let $\Sigma = \Sigma_1 = \Sigma_2 = \dots = \Sigma_k$. Then we may let $\theta = (\mu_1,\dots,\mu_k,\Sigma^{-1})$. The improper prior $g(\theta) = |\Sigma|^{(p+1)/2}$ is used. $Z_j|\theta,i_j$ is normal (μ_{i_j},Σ).

Single Allocation

$$f(z|X,i) \propto \left(\frac{N_i}{N_i+1}\right)^{p/2} \left[1 + \frac{N_i(x_i - z)^T S^{-1}(\bar{x}_i - z)}{(N_i+1)(N-k)}\right]^{-\frac{(N-k+1)}{2}}$$

Joint Allocation

$$f(z_1,\ldots,z_n|X,i_1,\ldots i_n)$$

$$\alpha\left(\prod_{i=1}^{k}\left(\frac{N_i}{N_i+n_i}\right)\right)^{p/2}\left|(N-k)S+\sum_1^k(M_i-\bar{x}_ie_i^T)\,\Omega_i\,(M_i-\bar{x}_ie_i^T)^T\right|^{-\frac{(N+n-k)}{2}}$$

where e_i is a $n_i \times 1$ column of ones and $\Omega_i = I - \dfrac{e_ie_i^T}{N_i+n_i}$.

Sequential allocation follows from the result for joint allocation.

Not Necessarily Equal Covariances

$$\theta = (\mu_1,\ldots,\mu_k,\Sigma_1^{-1},\ldots,\Sigma_k^{-1})$$

The improper prior $g(\theta)\,\alpha\,\prod_1^k|\Sigma\;|^{(p+1)/2}$ is used.

$$Z_j|\theta,i_j \quad \text{is normal } (\mu_{i_j},\ \Sigma_{i_j})$$

Single Allocation

$$f(z|X,i)\,\alpha\,\frac{\left(\frac{N_i}{N_i+1}\right)^{p/2}\Gamma\left(\frac{N_i}{2}\right)\left[1+\frac{N_i(\bar{x}_i-z)^TS_i^{-1}(\bar{x}_i-z)}{N_i^2-1}\right]^{-\frac{N_i}{2}}}{\Gamma\left(\frac{N_i-p}{2}\right)|(N_i-1)S_i|^{1/2}}$$

Joint Allocation

$$f(z_1,\ldots,z_n|X,i_1,\ldots,i_n)$$

$$\alpha\prod_{i=1}^{k}\left[(2\Pi)^{-\frac{pn_i}{2}}\frac{k^{-1}N_i-1,p}{k^{-1}N_i+n_i-1,p}\;\frac{|\Omega_i|^{p/2}\;|(N_i-1)S_i|^{\frac{N_i-1}{2}}}{\left|(N_i-1)S_i+(M_i-\bar{x}_ie_i^T)\Omega_i(M_i-\bar{x}_ie_i^T)^T\right|^{\frac{N_i+n_i-1}{2}}}\right]$$

where $$k_\upsilon^{-1},p = 2^{(\upsilon p)/2}\,\Pi^{\frac{p(p-1)}{4}}\prod_{i=1}^{p}\Gamma\left(\frac{\upsilon-i+1}{2}\right)$$

The Calculation for the Proceeding Case

$$L(X|\theta) \propto \prod_{i=1}^{k} |\Sigma_i^{-1}|^{\frac{N_i}{2}} \operatorname{etr}[-\tfrac{1}{2}\Sigma_i^{-1}[(N_i - 1)S_i + N_i(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T]]$$

so

$$p(\theta|X) \propto \prod_{i=1}^{k} |\Sigma_i^{-1}|^{\frac{N_i-p-1}{2}} \operatorname{etr}[-\tfrac{1}{2}\Sigma_i^{-1}[(N_i-1)S_i + N_i(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T]]$$

$$\propto \prod_{i=1}^{k} p(\mu_i, \Sigma_i^{-1}|X)$$

clearly $\mu_i|\Sigma_i^{-1}, X \sim N_p(\bar{x}_i, \Sigma_i/N_i)$

$$\Sigma_i^{-1}|X \sim \text{Wishart}(N_i-1, ((N_i-1)S_i)^{-1})$$

$$\int \prod_{j=1}^{n} f(z_j|\theta, i_j)\; p(\theta|X)\; d\theta$$

$$= \prod_{i=1}^{k} \int f(M_i|\mu_i, \Sigma_i^{-1})\; p(\mu_i, \Sigma_i^{-1}|X)\; d\mu_i\; d\Sigma_i^{-1}$$

$$f(M_i|\mu_i, \Sigma_i^{-1})$$

$$= (2\pi)^{-(n_i p)/2} |\Sigma_i|^{-n_i/2} \operatorname{etr}[-\tfrac{1}{2}\Sigma_i^{-1}(M_i - \mu_i e_i^T)(M_i - \mu_i e_i^T)^T]$$

$$p(\mu_i, \Sigma_i^{-1}|X)$$

$$= (2\pi)^{-p/2} \left|\frac{\Sigma_i}{N_i}\right|^{-\frac{1}{2}} \operatorname{etr}[-\tfrac{1}{2}(\Sigma_i/N_i)^{-1}(\mu - \bar{x}_i)(\mu - \bar{x}_i)^T]$$

$$\cdot\; k_{N_i-1,p}^{-1} |(N_i-1)S_i|^{(N_i-1)/2} |\Sigma_i^{-1}|^{(N_i-p-2)/2} \operatorname{etr}[-\tfrac{1}{2}\Sigma_i^{-1}(N_i-1)S_i]$$

where e_i^T is a row vector of n_i ones and

$$k_{v,p}^{-1} = 2^{(vp)/2}\, \pi^{\frac{p(p-1)}{4}} \prod_{i=1}^{p} \Gamma\left(\frac{v-i+1}{2}\right)$$

$$\int f(M_i|\mu_i,\Sigma_i^{-1})\, p(\mu_i,\Sigma_i^{-1}|X)\, d\mu_i$$

$$= k^{-1}_{N_i-1,p}\,|(N_i-1)S_i|^{(N_i-1)/2}\,|\Sigma_i^{-1}|^{(N_i-p-2)/2}\;\mathrm{etr}[-\tfrac{1}{2}\Sigma_i^{-1}((N_i-1)S_i)]$$

$$\cdot\,(2\Pi)^{-(pn_i)/2}\,|\Sigma_i|^{-n_i/2}\,N_i^{p/2}\int(2\Pi)^{p/2}\,|\Sigma_i|^{-\frac{1}{2}}$$

$$\mathrm{etr}[-\tfrac{1}{2}\Sigma_i^{-1}[(M_i-\mu_ie_i^T)(M_i-\mu_ie_i^T)^T+N_i(\mu_i-\bar{x}_i)(\mu_i-\bar{x}_i)^T]]\,d\mu_i.$$

Now note: $(M_i-\mu_ie_i^T)(M_i-\mu_ie_i^T)^T+N_i(\mu_i-\bar{x}_i)(\mu_i-\bar{x}_i)^T$

$$=(n_i+N_i)\left[\left(\mu-\frac{M_ie_i+N_i\bar{x}_i}{n_i+N_i}\right)\left(\mu-\frac{M_ie_i+N_i\bar{x}_i}{n_i+N_i}\right)^T\right]+(M_i-\bar{x}_ie_i^T)\,\Omega_i(M_i-\bar{x}_ie_i^T)^T$$

where $\Omega_i = I-\dfrac{e_ie_i^T}{n_i+N_i}$. Thus the above equals

$$k^{-1}_{N_i-1,p}\,|(N_i-1)S_i|^{(N_i-1)/2}\,|\Sigma_i^{-1}|^{(N_i-p-2)/2}\,\mathrm{etr}[-\tfrac{1}{2}\Sigma_i^{-1}((N_i-1)S_i)]$$

$$\cdot\left(\frac{N_i}{N_i+n_j}\right)^{p/2}(2\Pi)^{-(pn_i)/2}\,|\Sigma_i|^{-(n_i)/2}\,\mathrm{etr}[-\tfrac{1}{2}\Sigma_i^{-1}(M_i-\bar{x}_ie_i^T)\,\Omega_i(M_i-\bar{x}_ie_i^T)^T]$$

So finally, we compute

$$\left(\frac{N_i}{N_i+n_j}\right)^{p/2}(2\Pi)^{-(pn_i)/2}\;k^{-1}_{N_i-1,p}\;|(N_i-1)S_i|^{(N_i-1)/2}$$

$$\cdot\int|\Sigma_i^{-1}|^{\frac{N_i+n_i-p-2}{2}}\,\mathrm{etr}[-\tfrac{1}{2}\Sigma_i^{-1}[\,(N_i-1)S_i+(M_i-\bar{x}_ie_i^T)$$

$$\Omega_i\,(M_i-\bar{x}_ie_i^T)^T]]\,d\Sigma_i^{1}$$

recognizing the Wishart density, we obtain

$$\left(\frac{N_i}{N_i+n_i}\right)^{p/2}(2\Pi)^{-(pn_i)/2}\,|(N_i-1)S_i|^{(N_i-1)/2}\;\frac{k^{-1}_{N_i-1,p}}{k^{-1}_{N_i+n_i-1,p}}$$

$$\cdot\,|(N_i-1)S_i+(M_i-\bar{x}_ie_i^T)\,\Omega_i(M_i-\bar{x}_ie_i^T)^T|^{-(N_i+n_i-1)/2}$$

1.5 The Program

A program has been written to compute the above. The way to use the program is given below. The appendix is a listing of the program.

The program is written in 1977 standard Fortran. The sole exception is the program statement which, at the University of Minnesota, notifies the operating system which files (tape #s) are subject to READ and WRITE statements.

The Input File

The input file is Tape 2. The first record is the number of populations (np). This is an integer between 2 and 20 inclusive. Format is list directed.

The second input record is the procedure number. This is an integer between 1 and 6 inclusive. Format is list directed.

The procedures are numbered as follows:

Procedure 1: Single allocation, equal covariances.

Procedure 2: Joint allocation, equal covariances.

Procedure 3: Sequential allocation, equal covariances.

Procedure 4: Single allocation, possibly unequal covariances

Procedure 5: Joint allocation, possibly unequal covariances

Procedure 6: Sequential allocation, possibly unequal covariances

The third input record is a flag which should be 1 if the prior probabilities are known and 0 if the N_i are random and the parameter α_i of the Dirichlet prior are to be used. Format is list directed.

The next np records are to be such that the i^{th} is either the prior probability q_i of the i^{th} population or α_i. Format is list directed.

Next, the training sample is input. For each population the following set of records is required. The i^{th} set will refer to the i^{th} population.

For each population:

The first records consist of a label, the number of observations for that population and the dimension of the observation. The number of observations must be an integer between 1 and 50. The dimension must be an integer between 1 and 10. The next record is a character string giving the format for an observation. The format for these two records is (10X, 2I6/A80). Each subsequent record is an observation.

If the dimension varies from population to population execution is aborted.

Finally, the observations to be allocated are treated the same way as are the training sample observations of a particular population.

The Output

Procedures 1, 3, 5, 6 output the posterior probability for each of the populations.

In procedures 2 and 5 each possible outcome is represented by an allocation vector $\in \mathbb{R}^{np}$ the i^{th} element of which is the population to which the i^{th} observation is allocated. All possible allocation vectors are output with their corresponding posterior probabilities. Up to 1000 possiblities are allowed.

Example of an Input File and Corresponding Output

```
2
2
1
.5
.5
POP1             5     3 FORMAT
(       003G18.11      )
  3.0000000000       3.0000000000       2.0000000000
  4.0000000000       2.0000000000       3.0000000000
  4.0000000000       2.2000000000       3.1000000000
  2.4000000000       5.5600000000       3.3000000000
  1.4000000000       3.2000000000       2.4000000000
POP2             5     3 FORMAT
(       003G18.11      )
  8.9900000000       7.8900000000       9.2300000000
  6.8800000000       7.8590000000       9.9900000000
  8.8870000000       7.8890000000       9.5600000000
  5.8800000000       7.9870000000       6.8975000000
  8.9900000000       7.8958000000       9.8785000000
OBSNS            2     3 FORMAT
(       003G18.11      )
  5.5500000000       6.7800000000       4.2300000000
  4.4400000000       6.2500000000       5.8900000000
READY.
```

This file is for procedure 2 with 2 population and a known uniform prior.

ALLOCATION VECTOR	ALLOCATION VECTOR	ALLOCATION VECTOR	ALLOCATION VECTOR
1	1	2	2
1	2	1	2
POST PROB= .218	POST PROB= .268	POST PROB= .213	POST PROB= .302

ALLOCATION VECTOR WITH MAX POST PROB
2
2

POST PROB= .301961

IMSL

The following International Mathematical and Statistical Library Routines are used. The IMSL version mjst be compatible with 1977 standard Fortran.

BECOVM

GAMMA

LINV2P

VCVTFS

VIPRFF

VMULFP

VMULFS

See IMSL documentation.

2. EXAMPLES OF WIDE DISCREPANCIES FOR DIMENSION REDUCING OPTIMAL ALLOCATION AND SEPARATION PROCEDURES IN MULTIVARIATE NORMAL POPULATIONS.

2.1 Introduction

Suppose we have r populations the i^{th} of which is normal with mean $\mu_i \in \mathbb{R}^p$ and covariance Σ (i.e. $N_p(\mu_i,\Sigma)$). There are various ways to measure how distinguishable or separate the populations are from one another. We may wish to classify observations known to have come from one of the populations.

A large p may be troublesome. We ask: can we find a linear map from $\mathbb{R}^p$ to $\mathbb{R}^k$, $k<p$, such that the populations suffer a minimal loss of identity in the sense that some given measure of separation or the ease with which we classify diminishes minimally.

Examples are given below in which the linear map which is best according to the probability of correct classification (P.C.C.) is quite different from the optimal map obtained when the desire is to separate the populations from each other according to well known criteria.

2.2 A Simplifying Linear Map

The simplifying linear map is a composition of 3 simple linear maps. Let x be our response which originates from one of the r populations, the i^{th} of which is $N_p(\mu_i,\Sigma)$.

(i) $x \to z = \Sigma^{-\frac{1}{2}}x$. The i^{th} population is now $N_p(\delta_i, I_p)$ where $\delta_i = \Sigma^{-\frac{1}{2}}\mu_i$ and I_p is the population identity matrix.

(ii) Let $\bar{\delta} = \sum_{i=1}^{r} \delta_i/r$. Map $z \to y = z - \bar{\delta}$. The i^{th} population is now $N_p(\alpha_i, I_p)$ where $\alpha_i = \delta_i - \bar{\delta}$.

(iii) Let $A = [\alpha_1, \alpha_2, \ldots, \alpha_r]$. We use the singular value decomposition to write $A = V\Phi U^T$ where V is $p\times p$ orthogonal, U^T is $r\times r$ orthogonal and Φ is of the form

$$\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$$

where $D = \mathrm{diag}(d_1,\ldots,d_s)$ and $s = \mathrm{rank}(A)$. Without loss of generality we may assume $|d_1| \geqslant |d_2| \geqslant \ldots \geqslant |d_s|$. Finally we map $y \to u = V^T y$. Let $M = \Phi U^T = [\eta_1, \eta_2, \ldots, \eta_r]$. Our i^{th} population is now $N_p(\eta_i, I_p)$.

2.3 Reduction Without Loss

Given that the simplifying linear map has been applied, all of the p variables are now independent and the last $p - s$ are $N_1(0,1)$ in each population. Thus it is clear that for purposes of allocation and separation the last $p - s$ variables may be disregarded.

The simplifying linear map followed by a projection onto the space of the first s co-ordinates gives a linear map from $\mathbb{R}^p$ to $\mathbb{R}^s$ with no loss in P.C.C. or in any reasonable measure of separation.

2.4 Classification

Given the prior probability q_i that the observed x is from population i, the posterior probability that it is from population i is proportional to $q_i f(x|i)$ where $f(\cdot|i)$ is the density for the i^{th} population. An observed response is classified as having come from the population having the largest posterior probability. Let R_i be the subset of $\mathbb{R}^p$ such that if $x \in R_i$, x is allocated to population i.

The probability of correct classification is then

$$\sum_{i=1}^{r} q_i \int_{R_i} f(x|i)\, dx .$$

The P.C.C. can be considered to be one measure of how distinguishable the populations are from each other given the q_i's. In section 3 above we saw that we can map linearly from $\mathbb{R}^p$ to $\mathbb{R}^s$ without loss in P.C.C. Finding the optimal map to lower dimensions with P.C.C. as the criterion presents difficulties. (See Geisser (3) pages 304-309).

2.5 Separatory Criteria

Again our i^{th} population is $N_p(\mu_i,\Sigma)$. We consider separatory criteria (i.e. measures of how "far apart" the populations are) which may be expressed as a scalar function, increasing in and determined by, the roots of the matrix $\beta\Sigma^{-1}$ where $\beta = \sum_{i=1}^{r}(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T$ with $\bar{\mu} = \sum_{i=1}^{r} \mu_i/r$. This class includes well known criteria such as those due to Hotelling and Wilks.

We know we can reduce from p to s without loss. Given a criterion from the class defined above how can we reduce further with minimal loss of separation. The solution to this problem is known and is as follows.

We start by applying the simplifying linear map so that our populations are now $N_p(\eta_i, I_p)$ where $[\eta_1,\ldots,\eta_r] = M = \Phi U^T$ as in section 2. Consider matrices C mapping $\mathbb{R}^p$ to $\mathbb{R}^k$, $k < s$. For each C we look at the roots of the matrix

$$(CMM^TC^T)(CC^T)^{-1}$$

Since

$$MM^T = \begin{bmatrix} D \\ 0 \end{bmatrix}[D,0].$$

The roots of the above are also the roots of

$$[D.0]\, C^T(CC^T)^{-1}C\begin{bmatrix} D \\ 0 \end{bmatrix}.$$

The matrix $C^T(CCT)^{-1}C$ is idempotent so that (see Bellman, page 113) the i^{th} largest roots in less than the i^{th} largest roots of $[D,0]\begin{bmatrix} D \\ 0 \end{bmatrix}$ which is d_i^2. (recall $|d_1| \geqslant |d_2| \geqslant \ldots \geqslant |d_s|$). Since this bound is obtained by the choice $C_k^* = [I_k, 0]$ where I_k is the $k \times k$ identity matrix, this choice is optimal throughout the entire class of separatory criteria under consideration.

2.6 Examples

Given a separatory criterion from the class of section 5 above we know how to map linearly down to $\mathbb{R}^k$, $k < s$ with minimal loss. Given a set $\{q_i\}$, $i = 1,2,\ldots,r$ of prior probabilities we do not, in general, know how to map down to $\mathbb{R}^k$ with minimal loss of P.C.C.

The following examples show that one cannot, being ignorant of the optimal P.C.C. map, use the dimension reducing linear map optimal for the separatory criteria and be sure of being close to optimal in terms of P.C.C. Also it is clear in the examples below that the optimal linear map with respect to the separatory criteria does a terrible job of preserving the identities of the populations.

The examples are presented by giving the Φ and U^T of section 2 so that the i^{th} population is $N_p(\eta_i, I_p)$ where η_i is the i^{th} column of ΦU^T.

Since the separatory criteria treat all populations in the same manner it is reasonable to use $q_i = 1/r$ in evaluating the P.C.C. for purposes of comparison.

(i) Let $U_1 = (-1,-1,\ldots,-1,1,1,\ldots,1)/\sqrt{2n}$, where there are n minus ones and n plus ones. $r = 2n$ in this example and U_1 will be the first row of U^T. Let $U_2 \in \mathbb{R}^{2n}$ such that $U_2 \cdot U_1 = 0$, $U_2 \cdot e = 0$ where $e = (1,1,\ldots,1)$, $\|U_2\| = 1$, and there exist c_1, c_2 such that $c_1U_1 + c_2U_2 = (-n,-(n-1),\ldots,-1, 1,2,\ldots,n)$. Now let U^T be any orthogonal matrix with first row U_1, second row U_2, and all but the last row orthogonal to e. Finally let

$$\Phi = \begin{bmatrix} N & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & N-\varepsilon & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & 1 & & & & \cdot \\ \cdot & \cdot & & & \ddots & & 1 & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

The optimal linear map from $\mathbb{R}^p \to \mathbb{R}^k$ for $k=1$ for the criteria of section 5 was shown to be $C_1^* = (1,0,\ldots,0)$. After applying this map n of the populations are $N_1\left(-\frac{N}{\sqrt{2n}},1\right)$ and the other n are $N_1\left(\frac{N}{\sqrt{2n}},1\right)$. Each population has $(n-1)$ identical mates! This identity crisis is illustrated by the fact that the P.C.C. is less than $1/n$.

If, however, we use C proportional to $(c_1,c_2,0,\ldots,0)$ where the c_i are those above the P.C.C. is virtually one for large N and small ε. The vector of population means is approximately proportional to $N(-n,-(n-1),\ldots,-1,1,2,\ldots,n)$ for small ε. Thus for large N all of the populations are clearly identifiable.

(ii) In example (i) we considered $k=1$. Now we use the same idea to show that even if we let $k=2$ there may be a linear map to k which is clearly superior to C_2^* in terms of P.C.C. and in terms of preserving identity.

Let $U_1 = (-1,-1,\ldots,-1,1,\ldots,1)/\sqrt{2n}$ as in (i). Let $U_2 = (-1,-1,\ldots,-1,$ $n-1,1-n,1,1,\ldots,1)/\sqrt{2n(n-1)}$ where there are $(n-1)$ plus and minus ones. We have: $U_1 \cdot U_2 = U_1 \cdot e = U_2 \cdot e = 0$, $\|U_2\| = \|U_1\| = 1$. Let U_3 be any vector in $\mathbb{R}^{2n}$ such that $U_3 \cdot U_2 = U_3 \cdot U_1 = U_3 \cdot e = 0$, $\|U_3\| = 1$, and there exists $c_1,c_2,c_3 \in \mathbb{R}$ such that $c_1U_1 + c_2U_2 + c_3U_3 = (-n,-(n-1),-(n-2),\ldots,-1,1,2,3,\ldots,n)$. Now let U^T be any orthogonal matrix with first three rows U_1,U_2,U_3 in that order and such that all rows except the last are orthogonal to e.

Finally let

$$\Phi = \begin{bmatrix} N & 0 & . & . & . & . & . & 0 \\ 0 & N & 0 & . & . & . & . & 0 \\ . & 0 & N-\varepsilon & 0 & . & . & . & 0 \\ . & . & 0 & 1 & . & . & . & . \\ . & . & . & & 1 & & & . \\ . & . & . & & & . & & . \\ . & . & . & & & & 1 & . \\ 0 & 0 & 0 & . & . & . & . & 0 \end{bmatrix}$$

then

$$C_2^* = \begin{bmatrix} 1 & 0 & 0 & . & . & 0 \\ 0 & 1 & 0 & . & . & 0 \end{bmatrix}.$$

Using C_2^* the transformed populations have a P.C.C. of less than

$$\left(\frac{1}{2n}\right)(2) + \frac{1}{n}\left(\frac{1}{n-1}+\frac{1}{n-1}+\dots+\frac{1}{n-1}\right) = \frac{2}{n}$$

since all but 2 populations have (n-2) identical mates.

If, however, we use C proportional to $\overline{(c_1,c_2,c_3,0,\dots,0)}$ the transformed population have a P.C.C. which is virtually 1 for large N and small ε. Again, the populations are clearly identifieable for large N.

Clearly similar examples can be concocted for $k > 2$.

2.7 Comments

Given that the simplifying transformation has been performed, the matrices Φ and U^T give a complete set of parameters for the r populations.

The separatory criteria of section 5 depend only on Φ. The examples of section 6 show that any good criterion cannot ignore U^T.

The P.C.C., with uniform prior, appears to be a good measure but it requires distributional assumptions.

2.8 A Suggestion for Finding a Good Map to $\mathbb{R}$

Choose a vector $S = (s_1,\dots,s_r)$ which has "good spacing". If we have normality the best S would maximize the P.C.C. subject to $\|S\| = 1$ where the i^{th} population is $N_1(s_i,1)$. One could simply use $S = (-n,-(n-1),\dots,-1, 1,2,\dots,n)$ or some rearrangement of these numbers.

Apply the simplifying transformation.

Let U_i be the i^{th} row of U^T.

For any j such that $d_1^2, d_2^2, \ldots, d_j^2$ are all large relative to all of the d_i^2 (recall $d_1^2 \geqslant d_2^2 \geqslant \ldots \geqslant d_s^2$) project S onto the subspace spanned by $\{U_1, \ldots, U_j\}$. Let $\hat{S}$ be the projection and $\{c_1, c_2, \ldots, c_j\}$ such that $\hat{S} = \sum_1^j c_i U_i$. Let

$$\beta = \left(\frac{c_1}{d_1}, \frac{c_2}{d_2}, \ldots, \frac{c_j}{d_j}, 0, \ldots, 0\right) \in \mathbb{R}^p.$$

Finally, let $C^* = \beta / \|\beta\|$.

For a given S and j, the $c_1, c_2, \ldots, c_j$ depend only on U^T. $\|S - \hat{S}\|$ small is desireable and again a characteristic of U^T.

Given $\{c_1, c_2, \ldots, c_j\}$ $C^* \Phi U^T = \hat{S}/\|\beta\|$ thus a small $\|\beta\|$ is desirable. $\|\beta\|$ will be small given $\{c_1, c_2, \ldots, c_j\}$ if the d_i are large in absolute value.

In the above the roles of Φ and U^T are clearly separated. A good Φ will have large d_i^2. A good U^T will have small $\|S - \hat{S}\|$ for some j with large d_1^2, $d_2^2, \ldots, d_j^2$ and some S with good spacing.

A few examples of naturally generated data were examined but in each case the largest root so dominated the other that no gain in P.C.C. could be made. In this problems a look at the relative size of the first few ordered roots is a good way to see if the P.C.C. of C_1^* is possibly substantially less than the optimal P.C.C.

References

(1) Geisser, S. (1966). Predictive Discrimination from the book Multivariate Analysis, Academic Press.

(2) Geisser, S. (1982). Bayesian Discrimination from the book Handbook of Statistics, Vol. 2, P.R. Krishnaiah and L. Kanal eds, North Holland Publishing Co.

(3) Geisser, S. (1977). Discrimination, Allocatory and Separatory, Linear Aspects from the book Classification and Clustering, Academic Press.

(4) Bellman, R. (1960). Introduction to Matrix Analysis, McGraw-Hill, New York.

Appendix

```
LIST

 83/08/26. 14.46.52.
M77TS      PROGRAM   AL5

C MAIN PROGRAM
C X(I,J,K)=TRAINING SAMPLE,I TH VARIATE,J TH OBSERVATION
C            FROM THE K TH POPULATION
C A=POOLED COVARIANCE,SYMMETRIC STORAGE
C AP( ,I)=COVARIANC FOR THE I TH POPULATION SYMMETRIC STORAGE
C M(I,J)=MEAN OF THE I TH VARIATE,J TH POPULATION
C D=NUMBER OF VARIATES
C NP=THE NUMBER OF POPULATIONS
C SS(I)=THE SAMPLE SIZE OF THE I TH POPULATION
C N=THE TOTAL NUMBER OF OBSERVATIONS
C NN=THE NUMBER OF OBSERVATIONS TO BE ALLOCATED
C Z=OBSERVATION TO BE ALLOCATED,SINGLE ALLOCATION
C ZED(I,J)=I TH VARIATE OF THE J TH OBSERVATION TO BE ALLOCATED
C PRIOR(I)=EITHER THE PRIOR PROBABILITY OR ALPHA FOR THE ITH POP
C FLPR=1 IF THE PRIOR IS KNOWN,0 IF THE ALPHAS ARE INPUT FOR
C          THE DIRICHLET PRIOR
C
      PROGRAM ALLOC(OUTPUT,TAPE2,TAPE6=OUTPUT)
      INTEGER D,NP,SS(20),FLCONT,PROCN
      INTEGER N ,NN,FLPR,ALOC(20),INDEX
      REAL X(10,50,20),M(10,20)
      REAL A(55),AP(55,20),Z(10),ZED(10,20)
      REAL POSTPR(20),PRIOR(20),MAX
      COMMON /FILE1/ X
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      OPEN(2,FILE='TAPE2')
      REWIND 2
      CALL GETING(NP,2,20)
      CALL GETING(PROCN,1,6)
      CALL GPRI(NP,PRIOR,FLPR)
      CALL READFL
      CALL MNTS
      CALL GETOBS(PROCN,Z,ZED,NN,D)
      IF(PROCN.EQ.1) CALL PROC1(Z,POSTPR,PRIOR,FLPR)
      IF(PROCN.EQ.4) CALL PROC4(Z,POSTPR,PRIOR,FLPR)
      IF(PROCN.EQ.2) CALL PROC2(NN,ZED,PRIOR,FLPR,ALOC,MAX)
      IF(PROCN.EQ.5) CALL PROC5(NN,ZED,PRIOR,FLPR,ALOC,MAX)
      IF(PROCN.EQ.3) CALL PROC3(NN,ZED,POSTPR,FLPR,PRIOR)
      IF(PROCN.EQ.6) CALL PROC6(NN,ZED,POSTPR,FLPR,PRIOR)
      IF((PROCN.EQ.2).OR.(PROCN.EQ.5)) THEN
         PRINT *,/,'ALLOCATION VECTOR WITH MAX POST PROB'
         DO 35 I=1,NN
            PRINT *,ALOC(I)
```

```
 35        CONTINUE
           IF(MAX.GT.0) THEN
                PRINT *,/,'POST PROB=',MAX,/
           ENDIF
        ELSE
           INDEX=1
           DO 40 I=1,NP
              PRINT 300,/,I,POSTPR(I)
 300          FORMAT(1X,'POST PROB OF POP ',I3,'=',G11.3)
              IF(POSTPR(I).GT.POSTPR(INDEX)) INDEX=I
 40        CONTINUE
           PRINT 400,/,INDEX,POSTPR(INDEX)
 400       FORMAT(1X,'POP ',I3,' HAS MAX POST PROB OF ',G11.3)
        ENDIF
        END
C
C THIS FUNCTION COMPUTES V(TRANSPOSE)*M*V
C
       REAL FUNCTION QUAD(V,M,N)
        REAL V(10),M(55),TEMP(10),X
        INTEGER N
        CALL VMULSF(M,N,V,1,10,TEMP,10)
        CALL VIPRFF(V,TEMP,N,1,1,X)
        QUAD=X
        RETURN
        END
C
C THIS FUNTION COMPUTES PART OF THE WISHART CONSTANT
C
       REAL FUNCTION K(P,V)
        INTEGER P,V
        REAL PI
        PI=2.0*ASIN(1.0)
        K=((2.0)**(P*V/2.0))*(PI**(P*(P-1)/4.0))
        DO 5 J=1,P
           K=K*GAMMA((V+1.0-J)/2.0)
  5     CONTINUE
        RETURN
        END
C
C SINGLE ALLOCATION: GIVEN DENSITY EVALUATE POSTERIOR PROBABILITY
C
       SUBROUTINE PSTPR(F,NP,SS,PPR,PRI,FLPR)
        INTEGER NP,SS(20),MAX,FLPR
        REAL F(20),PRI(20),SUM,PPR(20)
        IF(FLPR.EQ.1) THEN
           DO 5 I=1,NP
              PPR(I)=F(I)*PRI(I)
  5        CONTINUE
        ELSE
           DO 10 I=1,NP
              PPR(I)=(SS(I)+PRI(I)+1.0)*F(I)
 10        CONTINUE
        ENDIF
        CALL NOR(PPR,NP)
        RETURN
        END
C
C IF FL=1 READ IN PRIORS,ELSE READ IN ALPHAS FOR DIRICHLET PRIOR
C
       SUBROUTINE GPRI(N,P,FL)
        INTEGER N,FL
        REAL P(20)
        CALL YESNO(FL)
        IF(FL.EQ.1) THEN
           CALL RDPRI(P,N)
```

```
      ELSE
         DO 5 I=1,N
            READ(Z,*) P(I)
  5      CONTINUE
      ENDIF
      RETURN
      END
C
C PROCEDURE 1:SINGLE ALLOCATION,SAME COVARIANCES
C
      SUBROUTINE PROC1(Z,PPR,PRI,FLPR)
       INTEGER D,NP,SS(20),N,IDGT,IER,FLPR
      REAL M(10,20),A(55),Z(10),AINV(55),DIF(10),F(20),
+     WKAREA(75),X,D1,D2,PPR(20),PRI(20),AP(55,20)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      IDGT=0
      CALL LINV2P(A,D,AINV,IDGT,D1,D2,WKAREA,IER)
      DO 5 I=1,NP
         DO 10 J=1,D
            DIF(J)=M(J,I)-Z(J)
 10      CONTINUE
         X=QUAD(DIF,AINV,D)
         F(I)=((SS(I)/(SS(I)+1.0))**(D/2.0))*
+ ((1.0+SS(I)*X/((SS(I)+1.0)*(N-NP)))**((NP-1-N)/2.0))
  5 CONTINUE
      CALL PSTPR(F,NP,SS,PPR,PRI,FLPR)
      RETURN
      END
C
C PROCEDURE 4:SINGLE ALLOCATION,POSSIBLY DIFFERENT COVARIANCES
C
      SUBROUTINE PROC4(Z,PPR,PRI,FLPR)
       INTEGER D,NP,SS(20),IDGT,IER(3),FLPR,N
      REAL AP(55,20),Z(10),M(10,20),TEMP(55),PPR(20)
      REAL DIF(10),F(20),WKAREA(75),X,TINV(55),PRI(20)
      REAL G1,G2,A(55)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      DO 5 I=1,NP
         DO 10 J=1,D
            DIF(J)=M(J,I)-Z(J)
 10      CONTINUE
         DO 15 J=1,D*(D+1)/2
            TEMP(J)=AP(J,I)
 15      CONTINUE
         CALL LINV2P(TEMP,D,TINV,IDGT,D1,D2,WKAREA,IER)
         X=QUAD(DIF,TINV,D)
         G1=GAMMA(SS(I)/2.0)
         G2=GAMMA((SS(I)-D)/2.0)
         F(I)=((SS(I)/(SS(I)+1.0))**(D/2.0))
+*G1*((1.0+SS(I)*X/(SS(I)**2-1.0))
+**(SS(I)/(-2.0)))/(G2*(D1*(2.0**D2)*((SS(I)-1.0)**D))**(.5))
  5 CONTINUE
      CALL PSTPR(F,NP,SS,PPR,PRI,FLPR)
      RETURN
      END
C
C A*(I + EE(TRANSPOSE)/NUM)*A(TRANSPOSE) WHERE E=(1,1,1....1)(TRANS)
C
      SUBROUTINE QUAD2(A,NUM,NR,NC,RES)
      REAL A(10,20),RES(55),MEG(210),TEM(10,20),TEMP(10,10)
       INTEGER NUM,NR,NC,NR1,NC1
      NC1=NC
      NR1=NR
      II=NC*(NC+1)/2
```

```
      DO 5 I=1,II
         MEG(I)=1.0/NUM
    5 CONTINUE
      CALL VMULFS(A,MEG,NR1,NC1,10,TEM,10)
      DO 10 I=1,NR1
         DO 15 J=1,NC1
            TEM(I,J)=TEM(I,J)+A(I,J)
   15    CONTINUE
   10 CONTINUE
      CALL VMULFP(TEM,A,NR1,NC1,NR1,10,10,TEMP,10,IER)
      CALL VCVTFS(TEMP,NR1,10,RES)
      RETURN
      END
C
C COMPUTES PREDICTIVE DENSITY (F) AT OBSERVATIONS TO BE ALLOCATED
C (ZED) GIVEN ALLOCATION PI,SAME COVARIANCES
C
     SUBROUTINE SCJTF(NN,ZED,PI,LN,F)
      INTEGER D,NP,SS(20),N,NN,PI(20),LN(20),IER,IDGT
      REAL M(10,20),A(55),ZED(10,20),F,DIF(10,20)
      REAL AP(55,20),TEMP1(55),TEMP2(55)
      REAL WKAREA(75)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      DO 2 I=1,NP
         LN(I)=0
    2 CONTINUE
      DO 3 I=1,D*(D+1)/2
         TEMP1(I)=0.0
    3 CONTINUE
      DO 5 I=1,NP
         DO 10 J=1,NN
            IF(PI(J).EQ.I) THEN
               LN(I)=LN(I)+1
               DO 20 K=1,D
                  DIF(K,LN(I))=ZED(K,J)-M(K,I)
   20          CONTINUE
            ENDIF
   10    CONTINUE
         IF(LN(I).NE.0) THEN
          CALL QUAD2(DIF,-(SS(I)+LN(I)),D,LN(I),TEMP2)
          DO 40 J=1,D*(D+1)/2
             TEMP1(J)=TEMP1(J)+TEMP2(J)
   40     CONTINUE
         ENDIF
    5 CONTINUE
      DO 55 I=1,D*(D+1)/2
         TEMP1(I)=TEMP1(I)+A(I)*(N-NP)
   55 CONTINUE
      CALL LINV2P(TEMP1,D,TEMP2,IDGT,D1,D2,WKAREA,IER)
      F=(D1*(2.0**D2))**((NN-NP-N)/2.0)
      DO 60 I=1,NP
         F=F*((FLOAT(SS(I))/(SS(I)+LN(I)))**(D/2.0))
   60 CONTINUE
      RETURN
      END
C
C DETERMINENTAL DENSITY,CALLED ONLY BY DCJTF
C
     REAL FUNCTION DETDEN(Y,NGAM,A,MM,M,P)
      REAL Y(10,20),A(55),DIF(10,20),TEMP(55),X,D1,D2,K,WKAREA(75)
      INTEGER NGAM,MM,M,P,IDGT,IER
      IF(M.GT.0) THEN
      PI=2.0*ASIN(1.0)
      CALL LINV2P(A,P,TEMP,IDGT,D1,D2,WKAREA,IER)
      DETDEN=((D1*2.0**D2)*(MM**P))**(MM/2.0)
```

```
      DETDEN=DETDEN*(1.0+FLOAT(M)/NGAM)**(P/2.0)
      CALL QUADZ(Y,NGAM,P,M,TEMP)
      DO 20 I=1,P*(P+1)/2
         TEMP(I)=TEMP(I)+A(I)*MM
   20 CONTINUE
      CALL LINV2P(TEMP,P,A,IDGT,D1,D2,WKAREA,IER)
      X=(D1*2.0**D2)**((MM+M)/2.0)
      DETDEN=(DETDEN/X)*K(P,MM+M)/(K(P,MM)*(2.0*PI)
+**(P*M/2.0))
      ELSE
         DETDEN=1.0
      ENDIF
      RETURN
      END
C
C SAME AS SCJTF EXEPT COVARIANCES NEED NOT BE THE SAME
C
     SUBROUTINE DCJTF(NN,ZED,PI,LN,F)
      INTEGER D,N,NP,SS(20),NN,PI(20),LN(20)
      REAL M(10,20),AP(55,20),ZED(10,20),F,TEMP(55)
      REAL DIF(10,20),A(55)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      DO 5 I=1,NP
         LN(I)=0
    5 CONTINUE
      F=1.0
      DO 10 I=1,NP
         DO 20 J=1,NN
            IF(PI(J).EQ.I) THEN
               LN(I)=LN(I)+1
               DO 30 K=1,D
                  L=LN(I)
                  DIF(K,L)=ZED(K,J)-M(K,I)
   30          CONTINUE
            ENDIF
   20    CONTINUE
         DO 40 J=1,D*(D+1)/2
            TEMP(J)=AP(J,I)
   40    CONTINUE
         F=F*DETDEN(DIF,(-(SS(I)+LN(I))),TEMP,SS(I)-1,LN(I),D)
   10 CONTINUE
      RETURN
      END
C
C PROCEDURE 2:JOINT ALLOCATION,SAME COVARIANCES
C
     SUBROUTINE PROC2(NN,ZED,PRI,FLPR,AL,MAX)
      INTEGER D,NP,SS(20),N,NN,AL(20),LN(20)
      INTEGER IND(20),L,II,CNT,FLPR,NN1
      REAL M(10,20),A(55),ZED(10,20),MAX,PRI(20)
      REAL PPR,SUM,AP(55,20),TEMP(1000)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      NN1=NN
      CNT=NP**NN
      IF(CNT.GT.1000) THEN
         PRINT *,/,'TOO MANY POSSIBLE ALLOCATIONS'
         STOP
      ENDIF
      MAX=0
      DO 5 I=1,NN1
         IND(I)=1
    5 CONTINUE
      L=NN
      I=1
```

```
      SUM=0
      DO 10 II=1,CNT
         CALL SCJTF(NN,ZED,IND,LN,PPR)
         CALL PSTPR2(PPR,NP,SS,PRI,LN,FLPR)
         TEMP(II)=PPR
         SUM=SUM+PPR
         CALL CHIND(MAX,PPR,AL,IND,NN)
         CALL NEX(L,I,IND,NP,NN)
   10 CONTINUE
      CALL PALS(SUM,TEMP,NP,NN)
      MAX=MAX/SUM
      RETURN
      END
C
C PROCEDURE 5:JOINT ALLOCATION POSSIBLY DIFFERENT COVARIANCES
C
     SUBROUTINE PROC5(NN,ZED,PRI,FLPR,AL,MAX)
      INTEGER D,NP,SS(20),N,NN,AL(20),LN(20)
      INTEGER IND(20),L,II,CNT,FLPR,NN1
      REAL M(10,20),A(55),ZED(10,20),MAX,PRI(20)
      REAL PPR,SUM,AP(55,20),TEMP(1000)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      NN1=NN
      CNT=NP**NN
      IF(CNT.GT.1000) THEN
         PRINT *,/,'TOO MANY POSSIBLE ALLOCATIONS'
         STOP
      ENDIF
      MAX=0
      DO 5 I=1,NN1
         IND(I)=1
    5 CONTINUE
      L=NN
      I=1
      SUM=0
      DO 10 II=1,CNT
         CALL DCJTF(NN,ZED,IND,LN,PPR)
         CALL PSTPR2(PPR,NP,SS,PRI,LN,FLPR)
         TEMP(II)=PPR
         SUM=SUM+PPR
         CALL CHIND(MAX,PPR,AL,IND,NN)
         CALL NEX(L,I,IND,NP,NN)
   10 CONTINUE
      CALL PALS(SUM,TEMP,NP,NN)
      MAX=MAX/SUM
      RETURN
      END
C
C PROCEDURE 3:SEQUENTIAL ALLOCATION:EQUAL COVARIANCES
C
     SUBROUTINE PROC3(NN,ZED,PSTPR,FLPR,PRI)
      INTEGER D,NP,SS(20),N,NN,LN(20),L,IND(20),II,FLPR,NN1,NP1,CNT
      REAL M(10,20),A(55),ZED(10,20),PSTPR(20),PRI(20),PPR
      REAL AP(55,20)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      NN1=NN
      NP1=NP
      CNT=NP**NN
      DO 100 I=1,NP1
         PSTPR(I)=0.0
  100 CONTINUE
      DO 2 I=1,NN1
         IND(I)=1
    2 CONTINUE
```

```
      L=NN
      I=1
      DO 5 II=1,CNT
         CALL SCJTF(NN,ZED,IND,LN,PPR)
         CALL PSTPR2(PPR,NP,SS,PRI,LN,FLPR)
         PSTPR(IND(NN))=PSTPR(IND(NN))+PPR
         CALL NEX(L,I,IND,NP,NN)
    5 CONTINUE
      CALL NOR(PSTPR,NP)
      RETURN
      END
C
C GIVEN JOINT ALLOCATION PREDICTIVE DENSITY,THIS ROUTINE COMPUTES
C THE UNNORMALIZED POSTERIOR PROBABILITY
C
     SUBROUTINE PSTPR2(VAL,N,S,PRI,LN,FLPR)
      REAL VAL,PRI(20)
      INTEGER S(20),LN(20),FLPR,N
      IF(FLPR.EQ.1) THEN
         DO 5 I=1,N
            VAL=VAL*(PRI(I)**LN(I))
    5    CONTINUE
      ELSE
         DO 10 I=1,N
            DO 15 J=1,LN(I)
               VAL=VAL*(S(I)+PRI(I)+LN(I)-J+1)
   15       CONTINUE
   10    CONTINUE
      ENDIF
      RETURN
      END
C
C GENERATES ALLOCATION VECTORS
C
     SUBROUTINE NEXT(L,I,IND,NP,NN)
      INTEGER L,I,IND(20),NP,NN
   10 IF(L.NE.NN) GOTO 20
         I=1
   70    IF(I.GT.NP) GOTO 30
            IND(NN)=I
            RETURN
            ENTRY NEX(L,I,IND,NP,NN)
            I=I+1
         GOTO 70
C     ELSE
   20    IND(L)=1
   40    IF(IND(L).GT.NP) GOTO 30
            L=L+1
            GOTO 10
   60       IND(L)=IND(L)+1
         GOTO 40
   30 IF((L.EQ.1).AND.(IND(1).GT.NP)) GOTO 50
      L=L-1
      GOTO 60
   50 RETURN
      END
C
C PROCEDURE 6:SEQUENTIAL ALLOCATION,POSSIBLY DIFFERENT COVARIANCES
C
     SUBROUTINE PROC6(NN,ZED,PSTPR,FLPR,PRI)
      INTEGER D,NP,SS(20),NN,LN(20),L,IND(20),II,N,FLPR,NN1,NP1,CNT
      REAL M(10,20),AP(55,20),ZED(10,20),PSTPR(20),PRI(20),PPR
      REAL A(55)
      COMMON /FILE2/ D,NP,SS,N
      COMMON /FILE3/ M,A,AP
      NN1=NN
```

```
      NP1=NP
      CNT=NP**NN
      DO 100 I=1,NP1
         PSTPR(I)=0.0
  100 CONTINUE
      DO 2 I=1,NN1
         IND(I)=1
    2 CONTINUE
      L=NN
      I=1
      DO 5 II=1,CNT
         CALL  DCJTF(NN,ZED,IND,LN,PPR)
         CALL PSTPR2(PPR,NP,SS,PRI,LN,FLPR)
         PSTPR(IND(NN))=PSTPR(IND(NN))+PPR
         CALL NEX(L,I,IND,NP,NN)
    5 CONTINUE
      CALL NOR(PSTPR,NP)
      RETURN
      END
C
C KEEPS TRACK OF ALLOCATION VECTOR WITH LARGEST
C POSTERIOR PROBABILITY
C
     SUBROUTINE CHIND(OLD,NEW,INDEX,TEMP,N)
      REAL OLD,NEW
      INTEGER INDEX(20),TEMP(20),N,I
      IF(NEW.GT.OLD) THEN
         OLD=NEW
         DO 5 I=1,N
            INDEX(I)=TEMP(I)
    5    CONTINUE
      ENDIF
      RETURN
      END
C
C NORMALIZES A VECTOR
C
     SUBROUTINE NOR(VEC,N)
      INTEGER N
      REAL VEC(20),SUM
      SUM=0.0
      DO 5 I=1,N
         SUM=SUM+VEC(I)
    5 CONTINUE
      DO 10 I=1,N
         VEC(I)=VEC(I)/SUM
   10 CONTINUE
      RETURN
      END
C
C CALLS IMSL MOMENTS ROUTINE
C
     SUBROUTINE MOMENT(DIM,NOB,OBS,MEAN,COV)
      INTEGER DIM,NOB,NBR(6),IER
      REAL OBS(50,10),TEMP(10)
      REAL COV(55),MEAN(10)
      DATA TEMP /10*0.0/
      NBR(1)=DIM
      NBR(2)=NOB
      NBR(3)=NOB
      NBR(4)=1
      NBR(5)=0
      NBR(6)=0
      CALL BECOVM(OBS,30,NBR,TEMP,MEAN,COV,IER)
      RETURN
      END
```

```
C
C CALCULATES MOMENTS
C
      SUBROUTINE MNTS
       INTEGER D,NP,SS(20),N
       REAL X(10,50,20),M(10,20),A(55),AP(55,20)
       REAL TEMP(30,10),TEM(55),MN(10)
       COMMON /FILE1/ X
       COMMON /FILE2/ D,NP,SS,N
       COMMON  /FILE3/ M,A,AP
       DO 5 I=1,D*(D+1)/2
          A(I)=0
     5 CONTINUE
       N=0
       DO 10 I=1,NP
          N=N+SS(I)
          DO 15 J=1,SS(I)
             DO 20 K=1,D
                TEMP(J,K)=X(K,J,I)
    20       CONTINUE
    15    CONTINUE
          CALL MOMENT(D,SS(I),TEMP,MN,TEM)
          DO 25 J=1,D
             M(J,I)=MN(J)
    25    CONTINUE
          DO 30 J=1,D*(D+1)/2
             AP(J,I)=TEM(J)
             A(J)=A(J)+TEM(J)*(SS(I)-1)
    30    CONTINUE
    10 CONTINUE
       DO 50 I=1,D*(D+1)/2
          A(I)=A(I)/(N-NP)
    50 CONTINUE
       RETURN
       END
C
C READS AN INTEGER WHICH IS EXPECTED TO BE EITHER 1 OR 0
C
      SUBROUTINE YESNO(FLAG)
       INTEGER FLAG
       READ(2,*) FLAG
       IF((FLAG.NE.1).AND.(FLAG.NE.0)) PRINT *,
+         'WARNING 1 OR 0 EXPECTED'
       RETURN
       END
C
C READ AN INTEGER WHICH MUST BE BETWEEN B1 AND B2
C
      SUBROUTINE GETING(INT,B1,B2)
       INTEGER INT,B1,B2
       READ(2,*) INT
       IF((INT.LT.B1).OR.(INT.GT.B2)) THEN
          PRINT *,'INTEGER IS OUT OF BOUNDS, RUN ABORTED'
          STOP
       ENDIF
       RETURN
       END
C
C READS IN A PROBABLILTY BETWEEN 0 AND 1
C
      SUBROUTINE READP(P)
       REAL P
       READ(2,*) P
       IF((P.LT.0).OR.(P.GT.1)) PRINT *,
+         'WARNING: SHOULD HAVE 0<=P=<1'
       RETURN
```

```
      END
C
C READS IN THE PRIOR,PROBABILITIES MUST SUM TO 1
C
     SUBROUTINE RDPRI(P,N)
      INTEGER N
      REAL P(20),SUM
      SUM=0.0
      DO 5 I=1,N
         CALL READP(P(I))
         SUM= SUM+P(I)
    5 CONTINUE
      IF((SUM.GT.1.001).OR.(SUM.LT..99)) PRINT *,
+        'WARNING: PROBABILITIES SHOULD SUM TO ONE'
      RETURN
      END
C
C READ IN THE OBSERVATIONS TO BE ALLOCATED
C
     SUBROUTINE GETOBS(PRN,Z,ZED,NN,D)
      INTEGER PRN,NN,D,ND
      REAL Z(10),ZED(10,20)
      CHARACTER*80 IFOR
      READ(2,100) NN,ND,IFOR
  100 FORMAT(10X,2I6/A80)
      IF(ND.NE.D) THEN
         PRINT *,'DIMENSION MISMATCH, RUN ABORTED'
         STOP
      ENDIF
      IF((PRN.EQ.1).OR.(PRN.EQ.4)) THEN
         READ(2,IFOR) (Z(J),J=1,D)
      ELSE
         IF((NN.LT.2).OR.(NN.GT.20)) THEN
            PRINT *,/,'MUST HAVE 2<=# OF OBS TO BE ALLOCATED=<20'
            STOP
         ENDIF
         DO 5 I=1,NN
            READ(2,IFOR) (ZED(J,I),J=1,D)
    5    CONTINUE
      ENDIF
      RETURN
      END
C
C READS IN THE TRAINING SAMPLE,THE NUMBER OF
C VARIATES AND SAMPLE SIZES
C
     SUBROUTINE READFL
      INTEGER D,NP,SS(20),N,ND
      REAL X(10,50,20)
      CHARACTER*80 IFOR
      COMMON /FILE1/ X
      COMMON /FILE2/ D,NP,SS,N
      READ(2,100) SS(1),D,IFOR
  100 FORMAT(10X,2I6/A80)
      CALL EDIT(D,SS(1))
      DO 5 I=1,NP-1
         DO 10 J=1,SS(I)
            READ(2,IFOR) (X(K,J,I),K=1,D)
   10    CONTINUE
         READ(2,100) SS(I+1),ND,IFOR
         CALL EDIT(ND,SS(I+1))
         IF(ND.NE.D) THEN
            PRINT *,'DIMENSION MISMATCH, RUN ABORTED'
            STOP
         ENDIF
    5 CONTINUE
```

```
      DO 15 J=1,SS(NP)
         READ(2,IFOR) (X(K,J,NP),K=1,D)
   15 CONTINUE
      RETURN
      END
C
C THE NEXT TWO POCEDURES PRINT THE ALLOCATION VECTORS AND
C THEIR POSTERIOR PROBABILITES
C
     SUBROUTINE PALS(S,V,NP,NN)
      REAL S,V(1000),TEMP(4)
      INTEGER NP,CNT,NN,N,IND(20),NTEMP(20,4)
      N=NP**NN
      DO 5 J=1,NN
         IND(J)=1
    5 CONTINUE
      L=NN
      I=1
      CNT=0
  100 IF((CNT+1)*4.GT.N) GOTO 200
         DO 10 J=1,4
            TEMP(J)=V(4*CNT+J)/S
            DO 15 K=1,NN
               NTEMP(K,J)=IND(K)
   15       CONTINUE
            CALL NEX(L,I,IND,NP,NN)
   10    CONTINUE
         CALL PAL(4,NP,NTEMP,TEMP)
         CNT=CNT+1
         GOTO 100
  200 CONTINUE
      N=N-CNT*4
      IF(N.GT.0) THEN
      DO 20 J=1,N
         TEMP(J)=V(4*CNT+J)/S
         DO 25 K=1,NN
            NTEMP(K,J)=IND(K)
   25    CONTINUE
         IF(J.NE.N) CALL NEX(L,I,IND,NP,NN)
   20 CONTINUE
      CALL PAL(N,NP,NTEMP,TEMP)
      ENDIF
      RETURN
     END
     SUBROUTINE PAL(N,NP,NTEMP,TEMP)
      REAL TEMP(5)
      INTEGER N,NTEMP(20,5),NP
      CHARACTER*50 IFOR
      WRITE(6,100)
  100 FORMAT(//)
      WRITE(UNIT=IFOR,FMT=200) N
  200 FORMAT('(',I1,28H(5X,'ALLOCATION VECTOR',4X)))
      WRITE(6,IFOR)
      WRITE(UNIT=IFOR,FMT=300) N
  300 FORMAT('(',I1,13H(12X,I2,12X)))
      DO 5 I=1,NP
         WRITE(6,IFOR) (NTEMP(I,J),J=1,N)
    5 CONTINUE
      WRITE(UNIT=IFOR,FMT=400) N
  400 FORMAT('(/,',I1,28H('     POST PROB=',G12.3,1X)))
      WRITE(6,IFOR) (TEMP(I),I=1,N)
      RETURN
      END
     SUBROUTINE EDIT(DIM,SSZ)
      INTEGER DIM,SSZ
      IF((DIM.LT.1).OR.(DIM.GT.10)) THEN
```

```
      PRINT *,/,'MUST HAVE 1<=DIMENSION=<10'
      STOP
   ENDIF ,
   IF((SSZ.LT.1).OR.(SSZ.GT.50)) THEN
      PRINT *,/,'MUST HAVE 1<=POP SAMPLE SIZE=<50'
      STOP
   ENDIF
   RETURN
   END
READY.



BYE

EXDG046   LOG OFF     14.59.34.
AN1I020   SRU         0.500 UNTS.

OD▒w.\
```